

### Amendments to the Claims:

Please amend the claims as indicated.

1. (Currently Amended) A prioritization apparatus for data in a communication channel, comprising:

a prioritization module comprising executable code stored on a storage device, executed by a processor, and configured to define a plurality of prioritization levels;

a communication module comprising executable code stored on the storage device, executed by the processor, and configured to process tasks over a plurality of communication channels and indicate that tasks are processed on selected communication channels;

an upgrade module comprising executable code stored on the storage device, executed by the processor, and configured to upgrade the prioritization level of unsuccessful tasks; and

a task controller comprising executable code stored on the storage device, executed by the processor, and configured to increment a counter for a selected communication channel, decrement the counter if the task is processed, and if the task is unsuccessful, maintain system resources on [[a]]the failed selected target channel by maintaining a count of the counter while resubmitting the[an]] unsuccessful task to a different channel and decrementing the counter when the task is successfully processed on the different channel.

2. (Original) The prioritization apparatus of claim 1, wherein the task controller is further configured to communicate with a status module, the status module configured to indicate system resource usage of a target device operatively coupled to the channel.

3. (Original) The prioritization apparatus of claim 2, wherein the target device further comprises a computer readable storage device.

4. (Currently Amended) The prioritization apparatus of claim 1, wherein the task controller further comprises a load module comprising executable code stored on the storage device, executed by the processor, and configured to distribute tasks across the plurality of communication channels according to a load balancing scheme.

5. (Original) The prioritization apparatus of claim 4, wherein the load balancing scheme dedicates a majority of system resources to tasks with a high priority, and a minority of system resources to tasks with a lower priority.

6. (Original) The prioritization apparatus of claim 1, wherein the task controller further comprises a plurality of counters for each of the plurality of channels, the counters configured to track system resource usage of the plurality of channels.

7. (Currently Amended) A device controller apparatus, comprising:  
a prioritization module comprising executable code stored on a storage device, executed by a processor, and configured to define a plurality of prioritization levels;  
a communication module comprising executable code stored on the storage device, executed by the processor, and configured to process tasks over a plurality of communication channels and indicate that tasks are processed on selected communication channels;

an upgrade module comprising executable code stored on the storage device, executed by the processor, and configured to upgrade the prioritization level of unsuccessful tasks;

a status module comprising executable code stored on the storage device, executed by the processor, and configured to track system resources of a device; and

a task processing module comprising executable code stored on the storage device, executed by the processor, and configured to receive tasks with upgraded prioritization levels, increment a counter for a selected communication channel, decrement the counter if the task is processed, and if the task is unsuccessful, maintain system resources on the failed selected target channel by maintaining a count of the counter while resubmitting the unsuccessful task to a different channel and decrementing the counter when the task is successfully processed on the different channel.

8. (Original) The device controller apparatus of claim 7, wherein the task processing module is further configured to receive tasks of different priorities according to a predefined prioritization scheme.

9. (Original) The device controller apparatus of claim 7, further comprising a queue of tasks to be processed.

10. (Original) The device controller apparatus of claim 7, wherein the task processing module is further configured to place tasks with upgraded prioritization levels at the beginning of the queue for processing.

11. (Canceled)
12. (Currently Amended) The device controller apparatus of claim 7, wherein the task processing module is configured to release system resources after the failed task is successfully ~~processed/completed~~ on ~~[[a]]~~the different channel.
13. (Original) The apparatus of claim 7, further comprising a computer readable storage device coupled to the apparatus and configured to process read/write tasks received from the task controllers.
14. (Original) The apparatus of claim 13, wherein the computer readable storage device is configured to process input/output tasks from the plurality of task controllers.
15. (Currently Amended) A system for task prioritization, the system comprising:  
a data communications network comprising a plurality of communication channels;  
a target device coupled to the network, the target device configured to receive tasks over the network, the target device comprising a status module comprising executable code stored on a storage device, executed by a processor, and configured to track system resources;  
a server coupled to the network, the server configured to receive read/write tasks from a client device and transfer the task to the target device;  
a task controller coupled to the server and configured increment a counter for a selected communication channel, decrement the counter if the task is processed, and if the task is unsuccessful, to maintain system resources on ~~[[a]]~~the failed

selected target channel by maintaining a count of the counter while resubmitting the unsuccessful task to a different channel and decrementing the counter when the task is successfully processed on the different channel; and

an upgrade module comprising executable code stored on the storage device, executed by the processor, and operatively coupled to the server, the upgrade module configured to upgrade the prioritization level of [[an]]~~the~~ unsuccessful task and communicate the unsuccessful task from to [[a]]~~the~~ different channel.

16. (Currently Amended) The system of claim 15, further comprising a prioritization module comprising executable code stored on the storage device, executed by the processor, and coupled to the server and configured to define a plurality of prioritization levels.

17. (Currently Amended) The system of claim 15, wherein the ~~task controller further comprises a counter that~~ is updateable and configured to indicate system resource usage of the target device.

18. (Original) The system of claim 15, wherein the prioritization module is configured to allocate a majority of system resources to a task with a higher priority and a minority of system resources to a task with a lower priority.

19. (Currently Amended) A method for maintaining task prioritization and load balancing, the method comprising:

selecting a communication channel, processing a task over the selected communication channel, and updating a counter according to utilized system

resources by incrementing the counter for the selected communication channel and decrementing the counter if the task is processed;

upgrading a prioritization level of ~~[[an]]the unsuccessful~~-task and communicating the ~~unsuccessful~~-task to a different channel if the task is unsuccessful; and

maintaining system resources on ~~[[a]]the failed selected target channel by maintaining a count of the counter~~ while resubmitting ~~[[an]]the unsuccessful task to a second~~~~different~~ channel and decrementing the counter when the task is successfully processed on the second channel.

20. (Original) The method of claim 19, wherein selecting a communication channel comprises distributing tasks across the plurality of communication channels according to a load balancing scheme.

21. (Canceled)

22. (Currently Amended) The method of claim 19, further comprising incrementing a second counter on ~~[[a]]the second channel when processing the unsuccessful~~-~~failed~~ task on the second channel.

23. (Canceled)

24. (Canceled)

25. (Currently Amended) A computer readable storage medium comprising computer readable code configured to carry out a process for maintaining task prioritization and load balancing, the process comprising:

selecting a communication channel, processing a task over the selected communication channel, and updating a counter according to utilized system resources by incrementing the counter for the selected communication channel and decrementing the counter if the task is processed;

upgrading a prioritization level of ~~[[an]]the unsuccessful~~-task and communicating the ~~unsuccessful~~-task to a different channel if the task is unsuccessful; and

maintaining system resources on a failed target channel by maintaining a count of the counter while resubmitting ~~[[an]]the~~ unsuccessful task to a different channel and decrementing the counter when the task is successfully processed on the different channel.

26. (Currently Amended) The process of claim 25, wherein selecting ~~[[a]]the~~ communication channel comprises distributing tasks across the plurality of communication channels according to a load balancing scheme.

27. (Canceled)

28. (Canceled)

29. (Canceled)

30. (Currently Amended) A prioritization apparatus for data in a communication channel, comprising:

means for defining a plurality of prioritization levels;

means for upgrading the prioritization level of a[[n]] ~~unsuccessful~~-task and communicat[[e]]ing the ~~unsuccessful~~-task to a different channel if the task is unsuccessful;;

means for selecting a communication channel, processing a task over the selected communication channel, and updating a counter according to utilized system resources by incrementing the counter for the selected communication channel and decrementing the counter if the task is processed;

means for maintaining system resources on the failed selected target channel by maintaining a count of the counter while resubmitting the unsuccessful task to a different channel and decrementing the counter when the task is successfully processed on the different channel;

means for transmitting and receiving tasks over the plurality of communication channels; and

means for distributing tasks across the plurality of communication channels according to a load balancing scheme.